The Implementation of Blockchain in Banking System using Ethereum

Masum Bakaul
Department of CSE
Britannia University
Cumilla, Bangladesh

Nipa Rani Das
Department of CSE
Britannia University
Cumilla, Bangladesh

Madhabi Akter Moni
Department of CSE
Britannia University
Cumilla, Bangladesh

ABSTRACT

Our current banking system is based on a central server where every branch is connected to each other. If the server made any changes to the data of a branch then other branches get affected. In this system, Corruption can be easily occurred because of unauthorized access which is totally insecure in transaction systems. But, Blockchain is a secure system where the transactional history regarding crypto-currency cannot be modified or destructed. Since 2008, Blockchain has gained immense interest due to exclusion of third-party organization participation in monitoring of the transactions. Ethereum is a protocol which is based on Blockchain technology and has several benefits over other crypto-currency based system and is best suited for creating a secure lending system. Every Ethereum based system runs on 'Smart-Contracts' which are lines of code and makes the system automated. As the system gets automated, proper algorithms can make the system reliable and secure as each and every step of the system is maintained and executed by the algorithm inside the Smart-Contracts. Blockchain systems work with peer-to-peer networks and also uses a consensus algorithm that's why there is no possibility of data modification.

Keywords

Blockchain, Smart contract, Ethereum, Cryptocurrency, ETH Ether

1. INTRODUCTION

1.1 Overview

All the executed bitcoin transactions are considered to be a public ledger in the Blockchain technology. The records of transactions made in Bitcoin or other crypto-currency are stored in blocks and maintained across all the computers that are linked in a peer-to-peer network. A Blockchain is the "current" part of a Blockchain which records some or all of the recent transactions, and once completed, goes into the Blockchain as a permanent database. Each time a new block is generated based on the completion of each block. Blocks are linked together in a linear fashion where each block contains a hash value of the previous block. In comparison with the traditional banking systems, Blockchain keeps all the transaction histories. Chronological Bitcoin transactions are entered in a Blockchain which is similar to the regular transaction in the bank. Meanwhile, blocks are similar to individual bank statements. Blockchain keeps records of every Ethereum based transaction ever executed. Thus, it provides a relationship on past transactions that happened and also, generates values belonged to a particular address. Some developers have begun looking at the creation of other different Blockchain that allows for tradeoffs and improved scalability using alternative, completely independent Blockchain, thus, allowing for more innovation [14]. It secures the transactions in a way that any record of the transaction that occurred in the past, cannot be modified as the modification changes the hash of several blocks. Changing the hash of the blocks will inevitably result in the breakage of the links among the blocks. Also, all the peers connected to that system do not support that modification excluding the modifier, which is based on the consensus algorithm.

1.2 Background

Current banking systems are based on central server mechanisms where all the personal information of account holders, his/her bank balance, and all other necessary information related to the bank are stored. All other branches are connected to the central server where every branch retrieves personal information, bank balance and history from the server. Failure in the central server causes all other branches to fall down which results in great damage to its users.

1.2.1 Ethereum

Bitcoin provides a simple stack-based programming language known as Script, which can be used to create features such as multi-signatures and escrow transactions. Although this script can be considered to be a programming language, it is not turning completed and its capabilities are also limited. The main reason behind this is that a Turning complete Blockchain language can be easily taken advantage. A simple infinite loop can create a great complication in detecting which computations are needed to be performed. As an alternative to the Bitcoin system, Ethereum provides turning complete smart contracts that run on the EVM [1].

1.2.1.1 Ethereum Details

Ethereum is considered to be a state-transaction system. The objects in the Ethereum system are known as accounts. There are two main types of accounts: "externally owned accounts" and "contracts accounts." Externally owned accounts are controlled by private keys while contract accounts are fully autonomous and governed by their contract code. Both types of accounts have the ability to send messages and create new accounts. A message is essentially a transaction. Message transfer Ethers, which is the currency for Ethereum, from account to account. If a contract account receives a message, its code will be triggered in order to determine what actions to be executed next [1].

1.2.1.2 Ethereum Fees and Gas

Ethereum solution to the infinite loop/excessive computation problem" is to charge a small number of fees for the computation. Each message that is to be sent specifies an amount of Gas for the message. Each computational step requires some Gas. If the Gas is completely spent before the code is completely executed, the execution stops and all changes are reverted. Otherwise, if all the statements of a specific code-block are successfully executed, any leftover gas is returned to the sender of the message. This prevents infinite loops in any smart contracts, as the amount of Gas to be supplied for a transaction is finite, forcing all computation to come to an end eventually. This phenomenon also prevents malicious users from attacking the Ethereum network via excessive computation since the amount of computation is proportional to the amount paid. With the introduction of decentralized smart contracts, several protocols for decentralized lending have been proposed [1].

1.2.2 Siacoin

Sia is a cloud storage platform that is decentralized in nature [2]. Rather than renting cloud storage from a centralized platform, Sia users rent storage from each other. The User who provides storage is known as a host, agrees to store the client's data. The storage provider periodically provides proof of its continued storage. The storage provider is compensated for each proof they provide, but if they miss a proof, they are penalized. The proofs are publicly verifiable and are stored on the Blockchain [2]. Thus, the users need not verify each proof manually.

Table 1: Comparison of Ethereum and Siacoin [13]

Characteristics	ETH	Siacoin
Price (USD)	163.754	0.00156
Market Cap (USD)	17,896,695,830	65,271,269

The comparison between Siacoin and Ethereum demonstrates Siacoin cannot develop an application as it is an open source. It is also vulnerable to data theft and monitoring. Data mining into a block is also time-consuming in the case of Siacoin. On the other hand, Ethereum can easily develop an application and also highly efficient than the Siacoin transaction.

1.2.3 Monero

Monero is a cryptocurrency that is based on the CryptoNote protocol. CryptoNote provides users with an anonymous payment system using a technology named ring signature that allows users to sign messages on behalf of a group. Ring signatures obfuscate the transactions via mixins. Each output describes money that is waiting to be spent by its owner. Each input to a transaction references a list of previous outputs (the mixins), only one of which is actually being spent. Blockchain does not disclose the output of actual transactions that are being spent. Additionally, the ring signatures are linkable to prevent output from being spent twice. Monero provides users with persistent cryptographic identities called addresses. In order to hide the user identity, Monero uses Stealth Addresses to blind the address from Blockchain observers. In 2016, Noether et al. state Monero deployed ring-confidential transactions (ringCT), which hide the amount of money involved in a transaction. In 2017, Monero updated the ringCT protocol to decrease the memory size of the protocol [9].

Table 2: Comparison of Ethereum and Monero [13]

Characteristics	ETH	XMR
Price (USD)	163.816	65.835
Market Cap (USD)	17,903,463,062	1,145,607,327

The comparison found, Monero is unlink-able but Ethereum is linkable. Besides, Ethereum has smart contract technological features which have connected to the past transaction with present transaction. Monero is untraceable and the identification of any user, in particular, is lower to a minimum of all minimums. For that's why Ethereum is a much effective transaction than Monero.

1.2.4 Tumblebit

Tumblebit is a payment protocol that is used for anonymous payments. This Tumblebit protocol is fully compatible with Bitcoin. The Tumblebit protocol operates through a Tumbler that can be used to make payments that are not linkable. The importance of Tumblebit lies in the fact that the Tumbler is untrusted: The Tumbler can neither steal funds nor anonymize users [11].

1.3 Research Problem

In central server mechanisms, a hacker can easily modify the transaction records and hence personal information. As all the branches are connected to the central server then if the central server falls down, it will affect all the other branches. Corruption can be easily occurred due to unauthorized access. This mechanism is not secure and reliable for users.

1.4 Research Objectives

- Implementing Banking System in Blockchain using the Ethereum platform which is much secure and reliable than centralized mechanisms.
- There would be no central server, it will be based on the peer-to-peer networks, and so by the grace of consensus algorithm, no information and transaction record can be changed.
- In the central server system, if the server crashes, the system will be down, data may get lost, which affects all the branches. But in a peer-to-peer networks, there is no central server, every branch gets a copy of data and transaction records. If one computer goes down, others will not be affected.

2. LITERATURE REVIEW

In 2013, V. Buterin published the paper "Ethereum: A Next-Generation Smart Contract and Decentralized Platform" where the author described an alternative to the Bitcoin system, which (Ethereum) provides turning complete smart contracts that run on the EVM [1]. D. Vorick and L. Champine researched on "Sia: Simple Decentralized Storage" which published in November 2014 where the author explained about Siacoin transaction [2]. In 2013, N. V. Saberhagen published the paper "Monero: Crypto V2.0" in Whitepaper Database which is based on the CryptoNote protocol [9]. In 2017, S. Singh and N. Singh focused on "Blockchain: Future of financial and cybersecurity" which published in IEEE Xplore [6]. S. Nakamoto researched "Bitcoin: A Peer-to-Peer Electronic Cash System" where the researcher described the electronic cash system transaction and which is published in October 2008 [3]. In 2016, E. Heilman, L. AlShenibr, and F. Baldimtsi published the paper "TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub" where the author described TumbleBit [11]. S. Kulechov, R. Morano, and Q. Fang researched "ETHLend.io White Paper - Democratizing Lending" which is published in February 2018 [7]. In 2018, D. Vujicic, D. Jagodic, and S. Randiz published "Blockchain technology, bitcoin, and Ethereum: A brief overview," where the authors explained to the fundamental overview of Ethereum [10]. E. Schneider focused on “One ledger: Public Blockchain Whitepaper” which is published in June 2018 [12]. In 2018, G. Fenu, L. Marchesi, M. Marchesi, and R. Tonelli published “The ICO phenomenon and its relationships with Ethereum smart contract environment” where the researchers interpreted the relationship among Ethereum smart contract and ICO phenomenon [8].

3. METHODOLOGY

3.1 Implementation Tools

The following programming languages are used to build the proposed lending system:

- Solidity for writing Smart Contracts
- JavaScript for interacting with the Smart Contracts
- HTML for designing front-end

The following frameworks and libraries are used:

- React.js for developing front-end
- Node.js JavaScript Runtime Environment
- Mocha JavaScript Framework for testing purposes
- Web3 JavaScript Library Collection
- Truffle as Ethereum Virtual Machine (EVM)

The following browser extension is used:

- MetaMask for accessing Ethereum enabled distributed applications

3.2 Implementation Details

The proposed protocol is implemented by following the steps mentioned below:

- The Back-end of the proposed system consists of a smart contract. Starting from registration, depositing money, sending money, earning interests and all of these tasks will be managed by the contracts.
- 'Mocha' testing framework is used to test whether smart contracts are functioning properly.
- 'Solc' compiler is conducted to compile the contracts and JSON files are created corresponding to that smart contracts.
- The system is firstly implemented on Ethereum Test Network which is known as Rinkeby. To implement the system in the Rinkeby network, an account is created in Infura.com.
- After signing up, Infura.com provides a URL which is of Rinkeby Test Network. The byte codes and ABIs of the smart contracts located in JSON files are then deployed to the URL of the Rinkeby Test Network by Truffle-HD-Wallet-Provider and web3.
- After deploying the smart contracts to Rinkeby Test Network, the addresses of the smart contracts are returned and saved.
- An interactive front-end (GUI) is developed using React.js.
- The front-end of the proposed system interacts with the smart contracts using the contracts' addresses and byte code.
- Web3 works as the interface between the Rinkeby Test Network and Front-End. It creates a bridge between the test network and the implemented DAPP.
- MetaMask Extension is used to access the implemented Ethereum enabled DAPP from the browser.
- Managing the accounts and all the transactions are done by the MetaMask.

3.3 Solidity for Writing Back-End Logics

The following steps are used to build the proposed Back-End logics which are mentioned below:

- Three smart contracts are written using the Solidity programming language. Most of the algorithms of the proposed system are included in these contracts.
- The smart contract named ‘bankingSystem.sol’ contains four methods for each of the banking tasks.
- ‘Registration’ method is used to register Ethereum users into the p2p banking system.
- ‘Deposit Money’ are responsible for transferring Ethers from Ethereum users to his/her own bank account.
- ‘Send Money’ is used to send money from one’s own bank account to another user’s bank account.

3.4 Testing the contracts with Mocha

Before implementation of the smart contracts in the Ethereum networks, they need to be tested so that any bugs of these contracts will not cause any financial disasters to the users of the systems or other types of system crashes. These smart contracts are tested against different types of test cases. When a smart contract passes all the test cases, then it is considered as eligible to be uploaded on the test network. The test cases against which the three smart contracts are tested are the following:

- Registration of a new user.
- Taking 10 Ether from a newly registered users as an initial deposit.
- Debiting Ether into one’s account.
- Crediting Ether from his/her own account.
- Receiving interests from the bank.

3.5 Contracts Compilation using Solc

Before uploading the contracts to the Ethereum network, these contracts need to be compiled. Solc compiler is used to compile a smart contract and generates bytecode and an ABI for each of the contracts. This bytecode and ABI are uploaded to the Rinkeby Test Network and are required by the systems to interact with the smart contracts. All the bytecodes and ABIs of the proposed system's smart contracts are acquired by the compilation using Solc.

3.6 Accessing Rinkeby Test Network via Infura

Smart contracts are not directly implemented in the Main Network. Instead, contracts are first deployed to the Test Network so that any problems in the contracts can be identified and solved. After eliminating all the defects, the contracts are then uploaded to the main network. Infura is a website that makes it easier for the DAPP developers to access any type of test network and upload the smart contract there very easily. Signing up at Infura.com provides the facility of various test networks in which the developers can store their proposed system's smart contracts. The proposed system's smart contracts are uploaded to the Rinkeby test network by registering an account at Infura.com

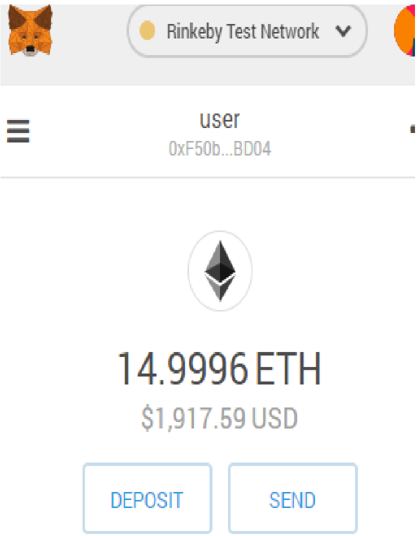
3.7 GUI Using React.js

React is one of the most popular framework of JavaScript which enables users to build interactive front-ends. The main reason behind using React.js for creating the GUI is that: React.js makes it easier to communicate with the Ethereum Network where our smart contracts are deployed. The user interface created by the React.js for the proposed system provides the following functionality:

1. Providing an interface for getting registered into the banking system. A text field will be available for inputting the amount of deposit money.
2. Balance Checking.
3. Receiving notification about interest.
4. Money transfer to other accounts.
5. Notification of receiving money from other accounts.
6. Cash Withdrawal.

4. RESULT

Figure 1, shows that a user has started the banking using Ethereum based Blockchain technology with 24 Ethers. The present system requires to expend at least 10 Ethers (fixed by authors in solidity framework) to create an account in the Bank. Therefore, the user has the remaining 14 Ether in his account.



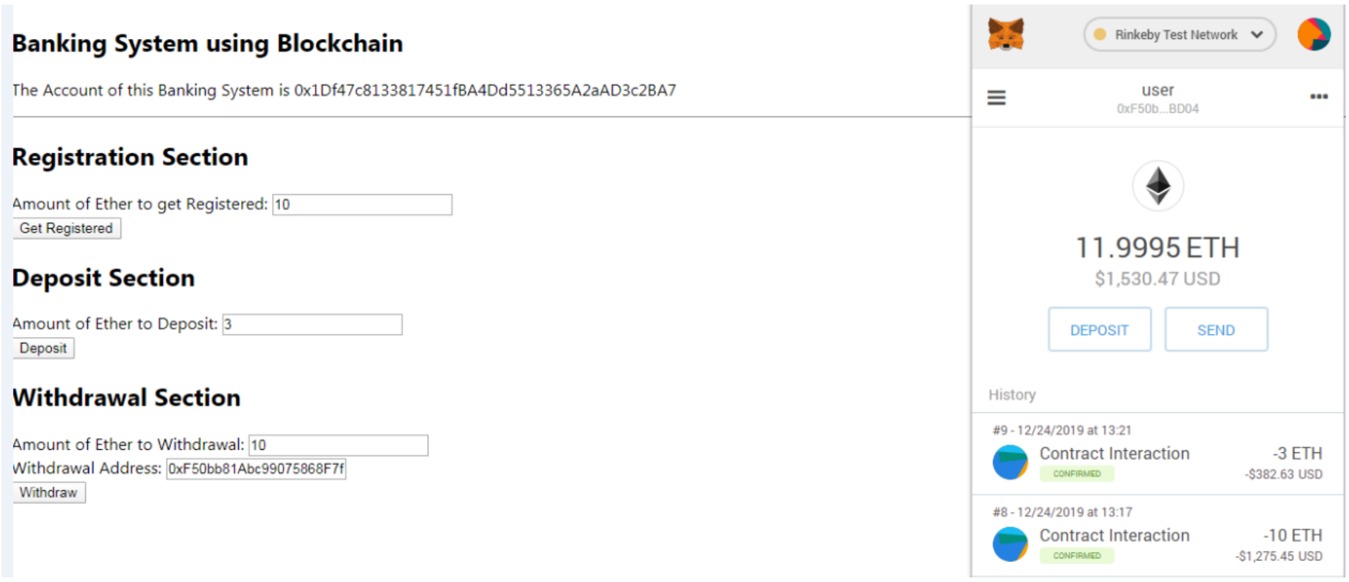
Rinkeby Test Network

user
0xF50b...BD04

14.9996 ETH
$1,917.59 USD

Figure 1: Getting registered using 10 Ether in Ethereum Based Banking System

An interactive GUI was design as shown in figure 2, which is used for user registration, Ether deposit, and for Ether withdrawal from the user account and developed through HTML, Solidity, JavaScript, and Reajct.js. Figure 2, shows that the particular user has deposited 3 Ethers from his account and now, the user has the remaining 11 Ethers in his account. The Rinkeby Test Network contains each and every transactional record that happened in the system given in figure 2.



Banking System using Blockchain

The Account of this Banking System is 0x1Df47c8133817451fBA4Dd5513365A2aAD3c2BA7

Registration Section

Amount of Ether to get Registered: 10

Deposit Section

Amount of Ether to Deposit: 3

Withdrawal Section

Amount of Ether to Withdrawal: 10
Withdrawal Address: 0xF50bb81Abc99075868F7f

Rinkeby Test Network

user
0xF50b...BD04

11.9995 ETH
$1,530.47 USD

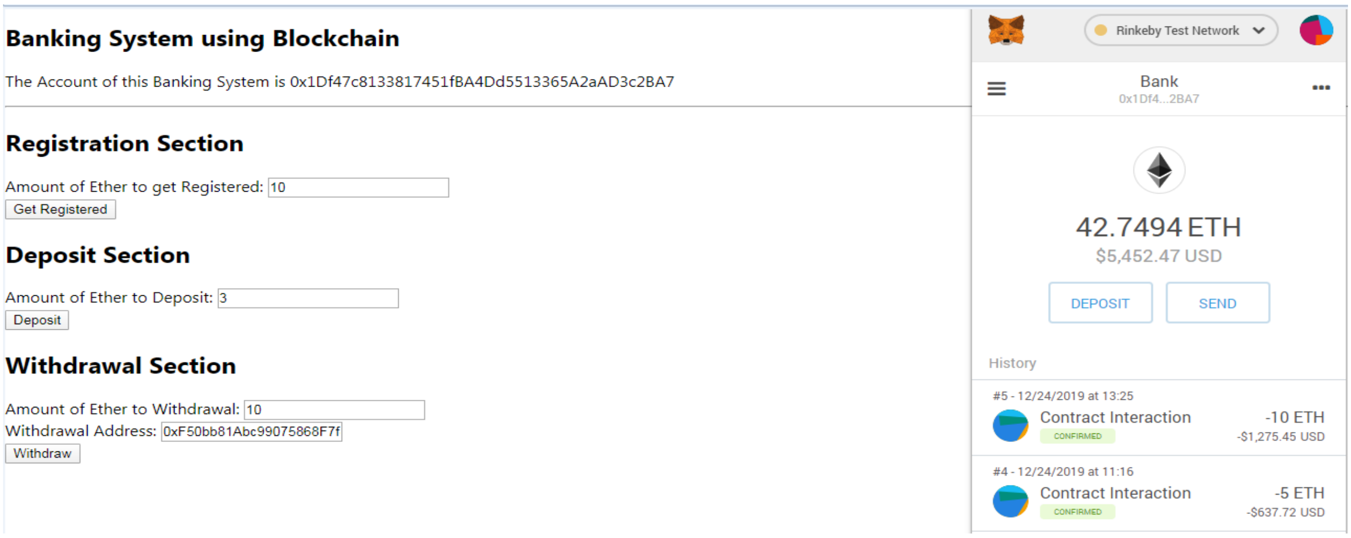
History

#9 - 12/24/2019 at 13:21
Contract Interaction CONFIRMED -3 ETH -$382.63 USD

#8 - 12/24/2019 at 13:17
Contract Interaction CONFIRMED -10 ETH -$1,275.45 USD

Figure 2: Depositing 3 Ether in Ethereum based Banking System.

In Figure 3, the user has withdrawn 10 Ethers from the Bank. Now, the user holds 21 Ethers as he revoked 10 Ethers from the Bank. The Rinkeby Test Network shown in figure 3 contains account information of the Bank. The consequent transactions will lead to reducing the number of Ethers stored in the bank.



Banking System using Blockchain

The Account of this Banking System is 0x1Df47c8133817451fBA4Dd5513365A2aAD3c2BA7

Registration Section

Amount of Ether to get Registered: 10

Get Registered

Deposit Section

Amount of Ether to Deposit: 3

Deposit

Withdrawal Section

Amount of Ether to Withdrawal: 10

Withdrawal Address: 0xF50bb81Abc99075868F7f

Withdraw

Rinkeby Test Network

Bank
0x1Df4...2BA7

42.7494 ETH
$5,452.47 USD

DEPOSIT SEND

History

#5 - 12/24/2019 at 13:25
Contract Interaction CONFIRMED -10 ETH -$1,275.45 USD

#4 - 12/24/2019 at 11:16
Contract Interaction CONFIRMED -5 ETH -$637.72 USD

Figure 3: Withdrawal of 10 Ether from Bank's vault to user account in Ethereum based Banking System

5. IMPLICATIONS

The Current Banking system is based on the central server mechanisms where all branches are connected to it. Any interruption to the central server will affect all other connected branches. On the other hand, Blockchain technology is a decentralized system. Moreover, the basic advantages of Blockchain technology are immutability, security, and transparency. The blockchain technology allows for verification without having to be dependent on third-parties. Businesses, in particular, could take advantage of this technology to reduce their costs and increase accountability. It uses protected cryptography to secure the data ledgers. Also, the current ledger is dependent on its adjacent completed block to complete the cryptography process. Since various consensus protocols are needed to validate the entry, it removes the risk of duplicate entry or fraud. Also, the transactions are recorded in chronological order. Thus, all the blocks in the Blockchain are time stamped. Since various consensus protocols are needed to validate the entry, it removes the risk of duplicate entry or fraud. Moreover, as the Blockchain system is implemented by Ethereum that's why it is very effective for the transaction system than the other transactions (i.e Siacoin, Monero, Bitcoin) because it has smart contract technological features which have connected to the past transaction with present transaction. Besides, Ethereum can easily develop an application as well as is public, open-source and Blockchain-based software platform that allows developers to build and deploy decentralized applications. It has no central point of failure, as it is being run from thousands of volunteer computers around the globe, which means it can never go offline.

6. FUTURE RECOMMENDATION

The facilities of getting a loan from the proposed banking system may be implemented and hence, will be included in the near future.

7. REFERENCES

[1] V. Buterin, "Ethereum: A Next Generation Smart Contract and Decentralized Platform," Github, Nov-2013.[Online].Available:https://gthub.com/ethereum/wiki/wiki/Ethereum-introduction

[2] D. Vorick and L. Champine, "Sia: Simple Decentralized Storage," Nebulous Inc, Nov. 2014.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," SantoshiNakamotoInstitute, Oct. 2008.

[4] E. Foundation, "Elastos: A Smart Web Powered by Blockchain White Paper IO," Whitepaper.IO, Jan. 2018.

[5] "Nexo:TheWorld's First Instant Cypto-backed Loans," Whitepaper Database, May 2018.

[6] S. Singh and N. Singh, "Blockchain: Future of financial and cyber security," IEEE Xplore, May 2017.

[7] S. Kulechov, R. Morano, and Q. Fang, "ETHLend.io White Paper - Democratizing Lending," Github, 25-Feb-2018. [Online]. Available: https://github.com/ETHLend/Documentation/blob/master/ETHLendWhitePaper.md.

[8] G. Fenu, L. Marchesi, M. Marchesi, and R. Tonelli, "The ICO phenomenon and its relationships with ethereum smart contract environment," IEEE Xplore, Mar. 2018.

[9] N. V. Saberhagen, "Monero: Crypto V2.0," Whitepaper Database, Oct. 2013.

[10] D. Vujicic, D. Jagodic, and S. Randic, "Blockchain technology, bitcoin, and Ethereum: A brief overview," *IEEE Xplore*, Apr. 2018.

[11] E. Heilman, L. AlShenibr, and F. Baldimtsi, "TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub," *Eprint Publication*, Jun. 2016

[12] E. Schneider, "Oneledger: Public Blockchain Whitepaper," *One Ledger*, Jun. 2018.

[13] "Ethereum Vs Monero Comparison - ETH/XMR Cryptocurrency Charts - 1 day," *Walletinvestor.com*. [Online]. Available: https://walletinvestor.com/compare/ethereum-vs-monero. [Accessed: 10-Jan-2020].

[14] K. Bheemaiah, "Block Chain 2.0: The Renaissance of Money," *LinkedIn*, 06-Jan-2015. [Online]. Available: https://www.linkedin.com/pulse/block-chain-20-renaissance-money-kariappa-bheemaiah.